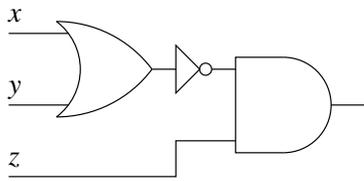


## Solutions, Exam 2, CSCI 210, Spring 2004

1. a.



b.  $\bar{x}z + \bar{y}z$

2. We can design the CD-ROM drive to send a hardware interrupt to the CPU when its response is ready. When a CPU receives a hardware interrupt, it will interrupt the currently running program and enter the operating system's interrupt handler, which can then handle the CD-ROM drive's response.

3. The only possibility is (d.).

```
4. public class Deadlock {
    Thread a;
    Thread b;

    public void runA() {
        synchronized(a) {
            synchronized(b) {
                System.out.println("I'm A");
            }
        }
    }
    public void runB() {
        synchronized(b) {
            synchronized(a) {
                System.out.println("I'm B");
            }
        }
    }
}
```

[In this example, a could get a lock on itself, and then a context-switch could allow b to get a lock on itself. When b then tries to get a lock on a, it will have to wait until a releases its own lock. And a will not be able to continue until b releases its lock. We have deadlock.]

5. There are two major reasons. Most importantly, programs using system calls will only work on operating systems supporting those system calls, whereas library functions will be supported by any platform supporting the language. Thus, programs using library functions are more portable.

Second, system calls tend to have very minimal functionality, because OS designers want to keep the OS as simple as possible to reduce the possibility of errors in the OS. The library functions, therefore, frequently allow a programmer to program without worrying about lower-level details. The difference in using `write` (which can display strings of characters only) and `printf` (which can translate other values into strings) is a good example of this.

```

6. public class Counter extends Thread { import java.io.*;
    public boolean stopped = false;    public class Main {
        public static void main(String[] args) {
            public void run() {
                int counter = 0;
                while(!stopped) {
                    ++counter;
                    System.out.println(counter);
                    try { Thread.sleep(1000); }
                    catch(InterruptedException e) {}
                }
            }
        }
    }
}

```

7.  $16 \times 10^{18}$